Amendment to the Claims:

The claims under examination in this application, including their current status and changes made in this paper, are respectfully presented.

1 (currently amended). A method for performing a write operation in a non-volatile memory system, the non-volatile memory system including an MLC NAND flash non-volatile memory having a plurality of erase units, each erase unit including a plurality of programming units arranged in a sequence, each programming unit comprising a plurality of non-volatile memory cells, wherein a first logical ~~unit~~ block is ~~associated with a first~~ mappable to one or more erase units and includes a plurality of logical ~~sub-units~~ pages arranged in a sequence, the method comprising:

writing data associated with the first logical block into an original erase unit to which the first logical block is mapped, so that data associated with the logical pages are written into corresponding programming units in that original erase unit in sequence;

grouping the programming units within ~~the~~ a first update erase unit into a plurality of groups of programming units, the programming units within each group arranged in a sequence comprising a first programming unit, a second programming unit, and a third programming unit;

writing contents of ~~a~~ an updated first logical ~~sub-unit~~ page into the first programming unit in a first group of programming units of the first update erase unit; ~~and~~

~~then~~ writing contents of ~~a~~ an updated third logical ~~sub-unit~~ page into the third programming unit of the first group of programming units, the third logical ~~sub-unit~~ page later in the sequence of logical sub-units than the first logical ~~sub-unit~~ page, and the third programming unit later in the sequence of programming units within the first group than the first programming unit and the second programming unit; ~~and~~

~~then writing~~ receiving contents of ~~a~~ an updated second logical ~~sub-unit~~ page of the first logical block ~~unit into a programming unit in a second group of programming units of the first erase unit~~, the second logical ~~sub-unit~~ page earlier in the sequence of logical ~~sub-units~~ pages than the third logical ~~sub-unit~~ page;

determining whether one of the groups of programming units in the first update erase unit is available for writing into;

responsive to a group of programming units being available in the first update erase unit, writing the updated contents of the second logical page into a programming unit in an available group of programming units of the first update erase unit; and

responsive to a second group of programming units not being available in the first update erase unit, merging contents associated with the first logical unit stored in the original erase unit and updated contents of the first logical block stored in the first update erase unit into a second update erase unit.

2 (canceled).

3 (currently amended). The method of claim 1, further comprising:

after the step of writing updated contents of a second logical ~~sub-unit~~ page, writing updated contents of a fourth logical ~~sub-unit~~ page into a programming unit of the second group, the fourth logical ~~sub-unit~~ page later in the sequence of logical sub-units than the second logical ~~sub-unit~~ page.

4 (canceled).

5 (currently amended). The method of claim 1 wherein each of the erase units is a physical block ~~and the first logical unit is a logical block~~;

and wherein each of the programming units is a physical page~~;~~

~~and wherein each of the logical sub-units is a logical page~~.

Claims 6 – 9 (canceled).

10 (currently amended). The method of claim ~~1~~ 3 wherein the first logical ~~sub-unit~~ page is included in a first logical grouping within the first logical ~~unit~~ block and the ~~third~~ fourth logical ~~sub-unit~~ page is included in a second logical grouping within the first logical unit;

and wherein the method further comprises:

writing a first indicator into the first group of programming units that substantially identifies the first group of programming units as associated with the first logical grouping; and

writing a second indicator into the second group of programming units that ~~substantially~~ identifies the second group of programming units as associated with the second logical grouping.

Claims 11 – 14 (canceled).

15 (previously presented). The non-volatile memory system of claim 22, wherein the controlling means comprises:

code devices for performing the sequence of operations; and
a processor for accessing and executing the code devices.

Claims 16 – 18 (canceled).

19 (currently amended). The non-volatile memory system of claim 22 wherein each of the erase units is a physical block ~~and the first logical unit is a logical block~~;

wherein each of the programming units is a physical page~~;~~

~~and wherein each of the logical sub-units is a logical page~~.

Claims 20 – 21 (canceled).

22 (currently amended). A non-volatile memory system comprising:

an MLC NAND flash non-volatile memory having a plurality of erase units, each erase unit including a plurality of programming units arranged in a sequence, each programming unit comprising a plurality of non-volatile memory cells;~~, wherein a first~~

means for mapping each of a plurality of logical blocks ~~unit is associated~~ with ~~a first~~ corresponding erase units, each logical block including ~~and includes~~ a plurality of logical ~~units~~ pages; and

means for controlling the writing of contents into the non-volatile memory according to groups of programming units within the erase units, the programming units within each group arranged in sequence comprising a first programming unit, a second programming unit, and a third programming unit, according to a sequence of operations comprising:

writing data associated with a first logical block into an original erase unit to which the first logical block is mapped, so that data associated with the logical pages are written into corresponding programming units in that original erase unit in sequence;

writing contents of a first logical ~~sub-unit~~ page into the first programming unit in a first group of programming units of ~~the~~ a first update erase unit;

then writing contents of ~~a~~ an updated third logical ~~sub-unit~~ page into the third programming unit of the first group of programming units, the third logical ~~sub-unit~~ page later in ~~the~~ a sequence of logical ~~sub-unit~~ pages than the first logical ~~sub-unit~~ page, and the third programming unit later in the sequence of programming units than the first programming unit and the second programming unit; and

~~then writing~~ receiving contents of ~~a~~ an updated second logical ~~sub-unit~~ page of the first logical block ~~unit into a programming unit in a second group of programming units of the first erase unit~~, the second logical ~~sub-unit~~ page earlier in the sequence of logical ~~sub-units~~ pages than the third logical ~~sub-unit~~ page;

determining whether one of the groups of programming units in the first update erase unit is available for writing into;

responsive to a group of programming units being available in the first update erase unit, writing the updated contents of the second logical page into a programming unit in an available group of programming units of the first update erase unit; and

responsive to a second group of programming units not being available in the first update erase unit, merging contents associated with the first logical unit stored in the original erase unit and updated contents of the first logical block stored in the first update erase unit into a second update erase unit.

Claims 23 – 48 (canceled).

49 (currently amended). The method of claim 1, wherein the step of writing contents of the second logical ~~sub-unit~~ page writes the contents into a second programming unit in the

second group of programming units of the first update erase unit, the second programming unit later in ~~the~~ a sequence of programming units than a first programming unit in the second group, and earlier in the sequence of programming units than a third programming unit in the second group.

50 (currently amended). The system of claim 22, wherein the operation of writing contents of the second logical ~~sub-unit~~ page writes the contents into a second programming unit in the second group of programming units of the first update erase unit, the second programming unit later in ~~the~~ a sequence of programming units than a first programming unit in the second group, and earlier in the sequence of programming units than a third programming unit in the second group.